Understanding the Market Through Sentiment Analysis

Agam Singh
University of Massachusetts, Amherst
Computer Science 490A

Abstract

We establish a baseline on predicting market returns and market volatility using logistic regression. We use this model to understand how overfitting might limit model performance. To negate the overfitting effects found, we propose further research to be considered. We conclude our experiments by training a deep neural network to assess how much we have improved our from our benchmark.

1 Introduction

In academia and the world of mathematical finance, markets are often modeled as being completely efficient. The *Efficient Market Hypothesis* states that all asset prices are fair. In simpler terms, this view claims that all prices in any market reflect all information that is present in the world and there can be no arbitrages (Joshi). An implication of such a view is that it is impossible or near impossible to forecast market moves. This is because there are no inefficiencies present that we might be able to capitalize on according to this view. This view is also favorited by academics to relax otherwise harsh conditions in many financial models like the Black-Scholes model. Models like the Black-Scholes model are used to price derivatives by simulating the stock price as a stochastic process to understand the risk involved with a stock and discounting from that risk the time value of money. This helps establish a mathematical model by which derivatives can be priced efficiently. Such models are usually completely quantitative, meaning that they use price information and volume to model how an asset might perform in the future. These models also rely heavily on what the asset has already done, usually just naively assuming that the asset's returns and volatility will remain what they were historically, an assumption that is never true in practice. Incorporating information from the news into a forecast of stock returns and volatility would yield much more fairly priced derivatives and would be an excellent tool for any investment firm.

In our paper, we are interested in understanding the relationship between news headlines and (1) market returns, represented as the returns of the Dow Jones Industrial Average (DJIA) and (2)

market volatility, represented as the *future T-day realized volatility,* defined below. To more formally define our questions, we are interested in understanding:

(1) Can the sign of market returns be predicted using news headlines?

(2) Can the future T-day realized volatility be predicted using news headlines?

We are interested in modeling market volatility from new headlines because of the important role volatility plays in modeling risk when it comes to pricing derivatives and constructing portfolios.

Experiment 1

To model a relationship between news headlines and stock market returns, we will train a logistic regression model to establish a baseline followed by a multilayer neural network to see if we can better model the nonlinearities present in the data. We will train these machine learning models to learn a mapping from news headlines on day t to the direction (negative or positive) of market returns on day $t + 1$.

We define the returns, R_t, on day t to be:

$$R_t = \ln\left(\frac{S_t}{S_{t-1}}\right)$$

Where S_t is the stock price at time t and sign(R_t) represents the sign (either positive or negative) of DJIA's return on day t. A sign of negative will be labeled as 0 and a positive sign will be label ed 1, turning this into a binary classification problem.

A visual inspection of DJIA's closing price and returns over time helps us understand the complexity of the problem:

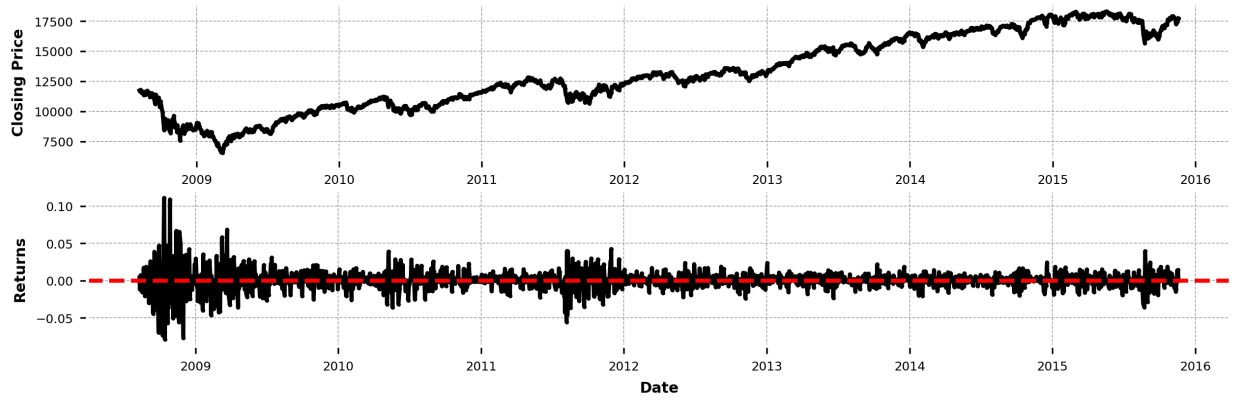


Figure 1: DJIA closing price and returns over time.

As we can see in figure 1, market returns are a mean reverting process with the mean being 0 and returns being normally distributed around the mean. Trying to learn a mapping between new headlines and stock returns is often written off by academics and quantitative finance groups as being "impossible". However, some researchers claim there is a statistically significant relationship as mentioned in the related work section.

Experiment 2

For our second experiment, we will try to construct a mapping from news headlines to market volatility. Since this is technically a regression problem, we will need to convert it to a binary classification problem. To convert it to a binary classification problem, we start by first computing the *future T-day realized volatility*. We define the future T-day realized volatility, $\sigma^2_{t+1:t+T}$, between day $t+1$ to $t+T$ as:

$$\sigma^2_{t+1:t+T} = \frac{252}{T}\sum_{i=1}^{T}\left[\ln\left(\frac{S_i}{S_{i-1}}\right)\right]^2$$

where S_i is the asset price at time i and T is our rolling window size.

For our experiment, we will try to forecast the future 14-day realized volatilities which can more or less be thought of as the standard deviation of the returns over the next 14 days from the day we are forecasting from. A high $\sigma^2_{t+1:t+T}$ will indicated heavy market movements and a lower $\sigma^2_{t+1:t+T}$ indicates relatively stable market conditions. A plot of the future 14-day realized volatilities exhibits some peculiarities:

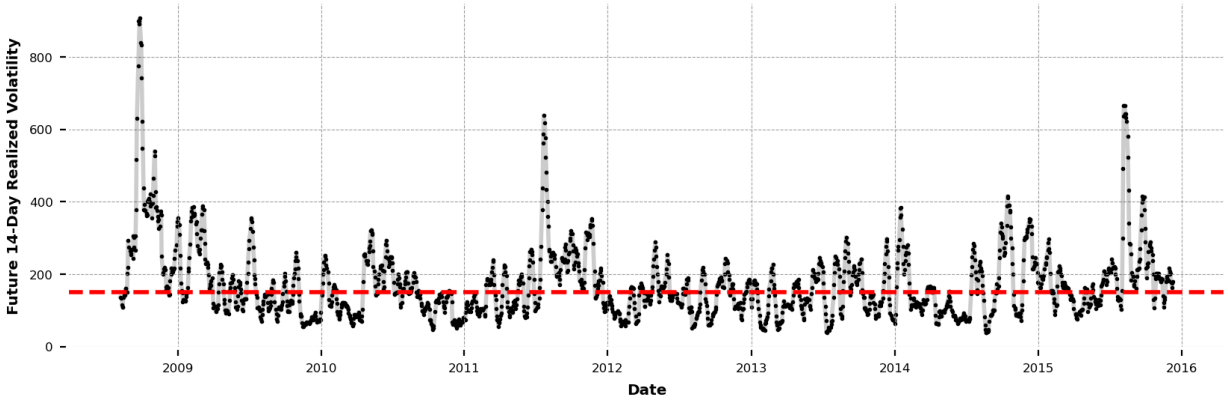


Figure 2: Future 14-Day Realized Volatilities over time (red line denotes $\bar{\sigma}^2$).

As we can see in figure 2, realized volatility is also a mean reverting process (Cont) with very pronounced periods of high market volatility (usually caused by global economic/political events). For example, the first major spike in figure 2 was a response to the 2008 financial crisis. This large spike in volatility might have been predicted from news headlines. The red line denotes the mean, $\bar{\sigma}^2$, across the entire time horizon. To ensure we get an even split of periods of high and low volatility, we propose the following method to cast volatility from a continuous set to a discrete, binary set:

- Compute the mean of all volatilities, $\bar{\sigma}^2$ over the entire time horizon.
- if $\sigma^2_{t+1:t+T} < \bar{\sigma}^2$, we cast it to 0 (low volatility).
- if $\sigma^2_{t+1:t+T} > \bar{\sigma}^2$, we denote that with a label of 1 (high volatility).

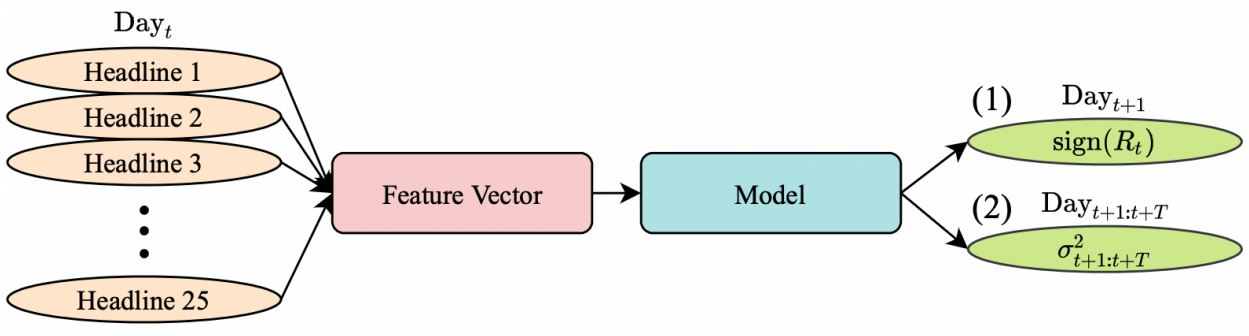


Figure 3: A mapping between news headlines and the sign of the returns and the future 14-day realized volatilities.

A General Framework

To construct models for such a problem, we propose a general framework by which we can run our experiments. As shown in figure 3, we first map the top 25 news headlines from any day into a feature vector. A model is then fitted or trained to predict (1) and (2) from these feature vectors.

Hypotheses

We hypothesize that forecasting market returns will not produce statically significant results. In other words, we expect the accuracy from any model we make for experiment 1 to be roughly 50% which is the accuracy we expect from randomly guessing labels. We suspect that forecasting market volatility will be much easier since it is usually very easy and intuitive for even humans to predict how the markets will be affected by factors like new headlines. It might be hard for humans to say whether or not the markets will go up or down in response to headlines but many would probably find it easy to say whether or not the markets will be "shaky" (volatile) in repose to certain news headlines. Further, realized volatility is often very easy to forecast in quantitative finance since exhibits many properties like clustering and mean reversion (Cont). We expect an accuracy significantly greater than 50% for experiment 2.

2 Related Work

While quantitative finance has been around for decades, applications of natural language processing in finance are still relatively new. And while hedge funds and firms employ many engineers to make creative models for efficiently pricing assets and forecasting risk, a lot of this work remains private to ensure they maintain a competitive edge with respect to the strategies they employ. With this in mind, we find that most work reaches the same conclusion: forecasting market returns is incredibly complex. Further, there is little work done on forecasting market volatility from news headlines. The work that does exist usually is centered on trying to forecast the direction of market returns (Joshi, Johan). This is usually a fruitless endeavor since market prices are best modeled as a stochastic process and if such an arbitrage existed, it would be priced in to the market very quickly. This is effect of "pricing in" is usually also called the no arbitrage condition (Hu).

3 Data

We are using the stocknews dataset from Kaggle:

https://www.kaggle.com/aaron7sun/stocknews

This dataset has 1988 rows and 27 total columns. The first column is the date, represented as a datetime object. The next is a label (either 0 or 1) to denote either positive or negative returns as described above, this is denoted as "label 1" in our dataset. The other 25 columns store the top 25 news headlines collected from reddit from that row's date. Note that the first headline is the top headline from that day, followed by the second most and so on. We joined this dataset with another dataset on the date column to add the closing of the DJIA for that date from which the volatiles could be computed. Once computed, we append a new column to denote the label for low and high volatility periods as described above. We denote this appended column as "label 2" to denote the labels for experiment 2. The columns of the final dataset are:

Date	Close	Volatility	Headline 1	…	Headline 25	Label 1	Label 2

Note: Label 1 represents the labels for our first research question while label 2 holds the labels for our second research question.

Finally, we provide some useful data statistics:

	Research question (1)	Research question (2)
Number of headlines	49,700	49,350
Number of rows in train set	1,590	1,582
Number of rows in test set	398	392
Number of columns	30	30

Note that the first 14 rows are excluded from dataset in experiment 2 since we can not calculate the future 14-day volatilities of those days.

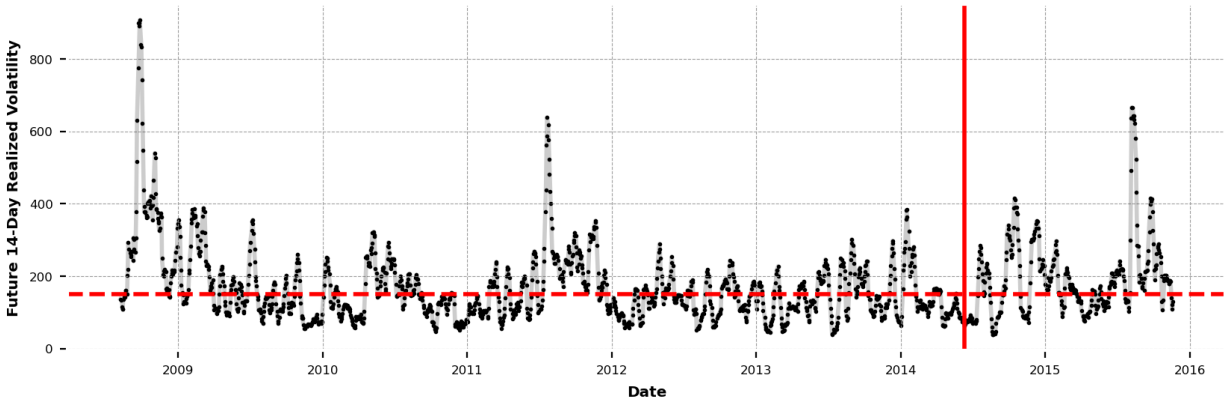
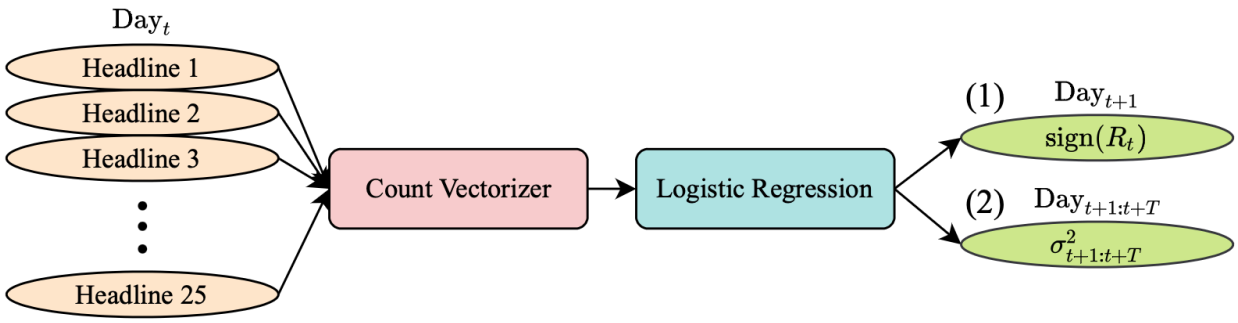


Figure 4: Defining a split between the train and test set (denoted by vertical red line).

The training and testing sets are constructed once the labels for both experiments have been computed. We use an 80-20 split as shown in figure 3. All the datapoints to the left of the vertical red line are in the training set and all the points to the right are in the test set.

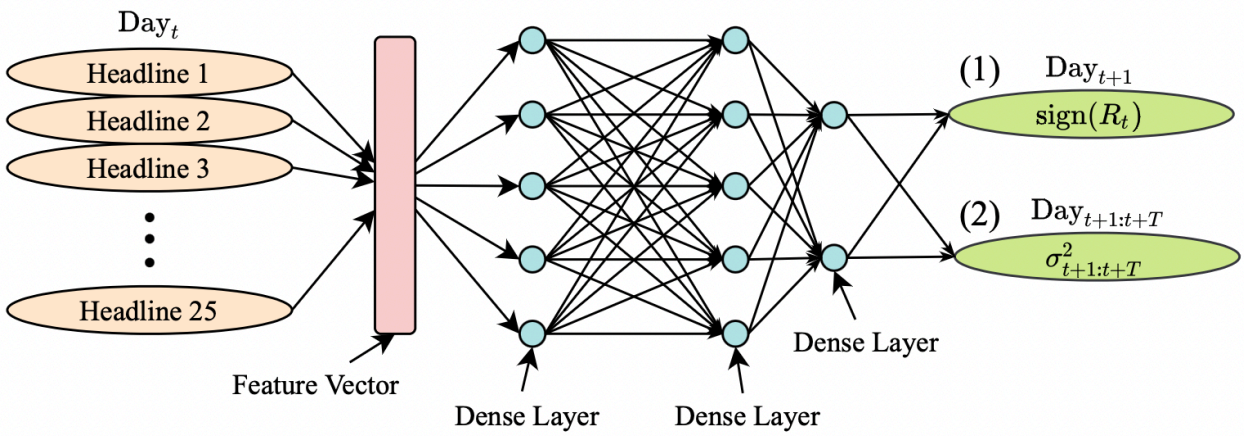
4 Method

For experiments 1 and 2, we wish to establish a baseline and we will do this using the logistic regression model with L2 regularization. To construct our feature vectors, we use a count vectorizer with an n-gram range from 1 to 3. We do this since we believe up to trigrams could be useful in modeling complex relationships between news headlines and returns/volatility. The following model represents the mapping from input to output:



Model 1: A baseline mapping between news headlines on day t and sign(R_t) and $\sigma^2_{t+1:t+T}$

Once a baseline has been established, we construct and train a deep neural network as we believe a neural network is needed to better learn the nonlinearities in the data. We suspect that the neural network might cause overfitting issues, to address this concern, we will add dropout layers between all the dense layers. We will also use linear activations in all layers except the last which will be a layer with two nodes (as seen in model 2) with softmax activation and select the more likely choice as our prediction (either 0 or 1 for experiment 1 and 2).



Model 2: A neural network mapping between news headlines on day t and sign(R_t) and $\sigma^2_{t+1:t+T}$

In model 2, to construct our feature vector, we first lowercase and collapse all the headlines from any given day into one string. We then tokenize the entire string. To do this, we use the keras tokenizer with the maximum number of words to keep set to 1,000. We then convert the texts to vectors using the tokenizers `texts_to_sequences` method which transforms the top 1,000 words from our text into a vector of integers. Once this has been done, we make sure all of our inputs have a length of 500 by padding the shorter vectors with zeros until they are of length 500.

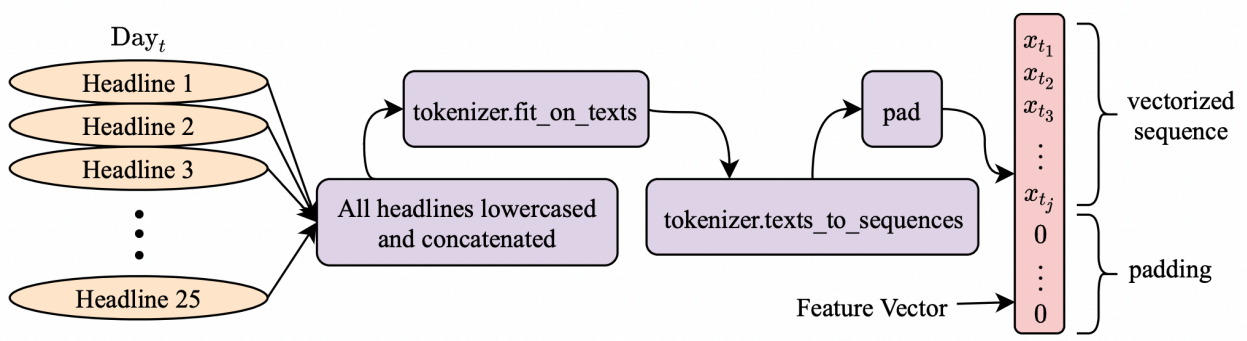


Figure 5: A visual description of vectorization procedure.

Once our feature vectors have been constructed, we train a multilayer neural network with 500 input nodes, 500 nodes in a hidden layer and 2 output nodes. A dropout layer with a 50% chance of dropout is added between the input and hidden layer and between the hidden and output layer.

The output layer consists of two nodes with softmax activation. This is interpreted as the probabilities we assign each class in both of our experiments. In the first experiment, the first output node represents the probability our model thinks the label is 0 (negative returns) and the second represents 1 (positive returns). We simply choose the value with the higher probability as our prediction for day *t*.

Similarly, for experiment 2, the first output node represents the probability our model thinks the label is 0 indicating the model thinks the next 14 days will exhibit low volatility and the second node represents the probability our model thinks the next 14 days will exhibit high volatility. Again, we choose the value with the higher probability as our prediction on the given input.

The neural network is trained with a batch size of 32 for 20 epochs.

5 Results

(1) Can the sign of market returns be predicted using news headlines?

We present the following results from models 1 and 2 on predicting market returns using news headlines from procedures defined above:

Model 1

		Predicted	
		0	1
Actual	0	43	153
	1	26	176

	Precision	Recall	F1-Score
0	0.62	0.22	0.32
1	0.53	0.87	0.66
Accuracy: 0.55			

Model 2

		Predicted	
		0	1
Actual	0	120	145
	1	52	75

	Precision	Recall	F1-Score
0	0.70	0.45	0.55
1	0.34	0.59	0.43
Accuracy: 0.50			

(2) Can the future 14-day realized volatility be predicted using news headlines?

We present the following results from models 1 and 2 on predicting 14-day realized volatility:

Model 1

		Predicted	
		0	1
Actual	0	81	146
	1	182	83

	Precision	Recall	F1-Score
0	0.31	0.64	0.42
1	0.64	0.31	0.42
Accuracy: 0.43			

Model 2

		Predicted	
		0	1
Actual	0	180	85
	1	75	52

	Precision	Recall	F1-Score
0	0.71	0.68	0.69
1	0.38	0.41	0.39
Accuracy: 0.59			

6 Discussion

As we can see from our results, predicting market returns is a very hard problem. New headlines are just one of many factors that impact market returns. Most of the factors that impact the markets might not even be measurable in any reasonable way. Quantitative, fundamental, technical and now sentiment analysts all prescribe what factors they believe hold the most weight

in shaping the markets. In reality, the problem requires a complex mix of *all* of these views, along with many others that we can't possibly model.

Many research papers discussing predictive models of stock returns boast accuracies even slightly above 50% as being "statistically significant" (Yun) since a model with true such capacity would be desirable by even the most competitive hedge funds on the planet. However, I believe that a slight nudge above 50% in backtests such as the ones performed in this paper are just caused by variance. In other words, if we conducted the same experiment on many different datasets, we would sometimes get accuracies above 50% and sometimes below. However, repeated sampling and experimentation would yield an *average accuracy* of 50%.

As far as understanding the market through natural language processing is concerned, it is a more worthwhile venture to model market volatility, which we attempted to do here. Our model 1 seems to do poorly, getting an accuracy of only 43%. Upon further investigation, we find that the features with the highest coefficients when trying to predict market volatility were words like:

{"oil spill", "wall street", "pakistan", "south africa", "hong kong", "vladimir putin", "china"}

This is clearly problematic since these are event that might have caused market volatility in the time horizon over the train set, but maybe they were events people largely ignored in the time horizon over the test set. So, these event might shake up the markets from 2009-2014 (time horizon of the train set), but perhaps not later. If our model learns that the terms "oil spill" cause market volatility, it might misclassify in later years when everyone has largely started ignoring the oil spill. Further, a more complex phenomenon might have occurred where the information associated with an oil spill was *priced in* to the market as per the Efficient Market Hypothesis.

To address this "overfitting" problem slightly, we constructed model 2 to better learn the non-linear and complex nature of the data. If we plot the data points correctly identified by model 2 over our test set, we see some interesting patterns:

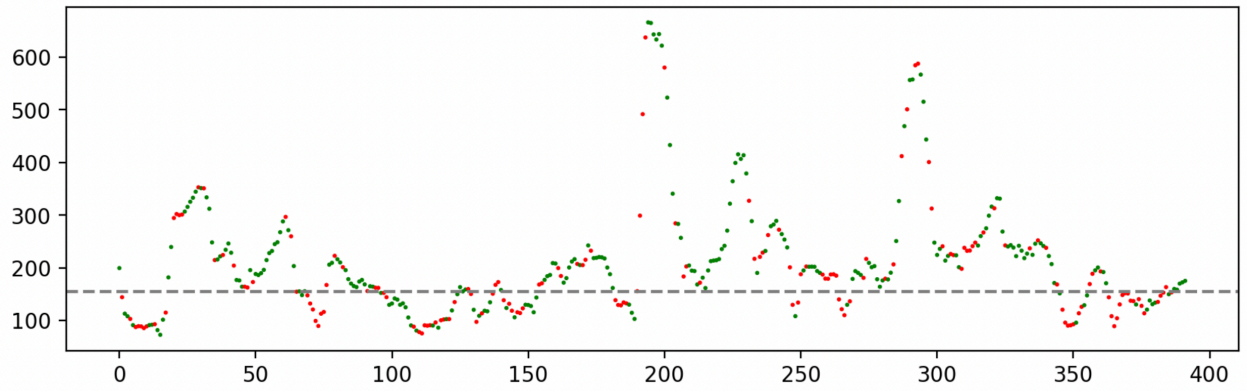


Figure 6: A visualization of realized volatility predictions by model 2.

In the above model, green points denote days correctly predicted by our model while red denotes wrong predictions. As we can see, our model seems to be predicting highly volatile days correctly but incorrectly predicting days with low volatility.

7 Future Work

There are many possibilities to explore with this problem. To address the overfitting problem identified in our market volatility experiment (where the model learns very specific events), perhaps we could used a named entity recognizer and a noun phrase extractor to replace specific names with broad categories. This way, our model will not overfit to headlines like “Donald Trump Fires Corey Lewandowski” which will have temporally local predicative power. Instead, perhaps we could train a model to learn “PERSON fires PERSON” is likely to cause market volatility in any year.

Further, we believe a lot more work could be done in the creation of feature vectors. Perhaps even the inclusion of important tokens from financial reports could help tremendously. We also might consider more specific applications of such a system. For example, trying to predict stock volatility for specific stocks like AAPL or TSLA from news headlines (or maybe subreddits and twitter hashtags) that contain information related to these stocks.

It is worth noting that to truly test such systems, especially with gradient decent based algorithms, many experiments should be conducted, and averaged statistics like accuracy should be used.

We believe much higher accuracies are possible for predicting market volatility in a binary classification setting, these will probably be made possible with a lot more creativity in how we express feature vectors and how we construct our models.

Citations

[1] Joshi, Kalyani & N, Bharathi & Rao, Jyothi. (2016). Stock Trend Prediction Using News Sentiment Analysis. International Journal of Computer Science and Information Technology. 8. 67-76. 10.5121/ijcsit.2016.8306.

[2] H. Yun, G. Sim and J. Seok, "Stock Prices Prediction using the Title of Newspaper Articles with Korean Natural Language Processing," 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2019, pp. 019-021, doi: 10.1109/ICAIIC.2019.8668996.

[3] Cont, Rama. "Volatility clustering in financial markets: empirical facts and agent-based models." *Long memory in economics*. Springer, Berlin, Heidelberg, 2007. 289-309.

[4] Johan Bollen, Huina Mao, Xiaojun Zeng, Twitter mood predicts the stock market, Journal of Computational Science, Volume 2, Issue 1, 2011, Pages 1-8, ISSN 1877-7503, https://doi.org/10.1016/j.jocs.2010.12.007.

[5] Hu, Hanlei, Zheng Yin, and Weipeng Yuan. "An Interval of no-Arbitrage Prices in Financial Markets with Volatility Uncertainty." *Mathematical Problems in Engineering* 2017 (2017)*ProQuest.* Web. 16 Dec. 2021.